UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Generation of synthetic data based on hidden Markov models

TREBALL FI DE GRAU

Grau en Enginyeria Informàtica

Autor: Jaime Ferrando Huertas

Tutores: Jorge Civera Saiz
Jens Lagersen

Curs 2017-2018

Final degree project

Generating synthetic data through Hidden Markov Models

Author:
Jaime Ferrando Huertas

Supervisors:
Jens Lagergren
Jorge Civera Saiz

Examiner:
Örjan Ekeberg (KTH)

Course:
Degree project in computer science, first cycle. DD142X

Home School: **Escuela Técnica Superior de Ingeniería Informática**
Home University: **UPV - Universitat Politècnica de València**
Host School: **School of computer science and communication**
Host University: **KTH - Royal Institute of Technology**
Date: **May 2018**

Abstract

Machine learning has becoming a trending topic in the last years, being now one of the most demanding careers in computer science. This growing has lead to more complex models capable of driving a car or cancer detection, however this models improvements are also thanks to the improvements in computational power. In this study we investigate a data exploration technique for creating synthetic data, a field of Machine learning that does not have as much improvements in the last years. Our project comes from a industrial process where data is a valuable asset, this process has both computational power and power full models but struggles with the availability of the data. In response for this a model for generating data is proposed, aiming to fill the lack of data during data exploration and training of this industrial process.

This model consist of a Hidden Markov Model where states represent different distributions the data follows, data is created by traveling through this states with an algorithm that uses the prior distribution of these states in a Dirichlet distribution.

The method to infer data distributions from the given data and create this Hidden Markov Model model has been explained along with the technique used to travel between states. Results have been presented showing how the data inferring performed and how the synthetic data reproduces the original one, taking special care for the reproduction of specific features in the original data. To get a better perspective of the data we created we tricked the states for our model, creating data from all of the states or from the states with less prior probability. Results showed that the model is capable of creating data similar to the real one but it struggled with data with a small amount of significant outliers. In conclusion a model to create reliable data have been introduced along with a list of possible improvements.

Sammanfattning

Maskininlärning har blivit ett populärt ämne de senaste åren, nu en av de mest krävande karriärvägarna inom datavetenskap. Att ämnet växt har lett till att mer komplexa modeller utvecklats, kapabla till exempelvis bilkörning och upptäckt av cancer. Dessa framgångar är dock också möjliga på grund av ökad beräkningskraft. I den här undersökningen undersöker vi ett område som utvecklats mindre jämfört med andra de senaste åren, data utforskning. En modell för att generera data föreslås, med målet att åtgärda bristen på data under datautforskning och träning. Denna modell består av ett HMM där tillstånd representerar olika fördelningar av dataflödet. Data skapas genom att färdas genom dessa tillstånd med en algoritm som använder a priori-fördelningen av dessa tillstånd i en Dirichlet-fördelning.

Metoden för inferens av datadistributionerna från den givna datan och därigenom skapa HMM modellen har förklarats tillsammans med tillvägagångssättet för att förflytta sig mellan tillstånd. Resultat har även presenterats som visar hur inferensen av datan presterade samt hur syntetisk data prestererade jämfört med den riktiga. För att få ett bättre perspektiv av datan vi skapat lurade vi tillstånden i vår modell, skapade data från alla tillstånden eller från tillstånden med lägre a priori sannolikhet. Resultaten visade att modellen är kapabel att skapa data lik den riktiga, men den hade svårt med data med en liten andel signifikanta outliers. Sammanfattningsvis så har en modell för att skapa pålitlig data introducerats tillsammans med en lista av möjliga förbättringar.

Contents

Chapter 1

Introduction

The last decade has seen a growing trend in Machine Learning, leading to the use of it in multiple fields. This growing has been thanks to the last years improvements in more complex models and computational power. In this paper we want to introduce and explore the idea of spending our resources in the actual data rather than in the more complex models or more power. Our proposal for data exploration is to create a model with the ability to recreate synthetic data from a given data collection. We also propose a use scenario where we believe this project could really make an impact.

Nowadays we count with both good computational power and capable models but our dataset exploration techniques have remained the same, we reached a point were spending our resources in improving models or computation do not report as much improvement as before. As stated this project wants to explore the capabilities of exploring the data, our work is based in the idea that exploring the data may lead to better results. The idea of focusing in the data rather that in models comes from several articles where they discuss the data importance[1] and the dataset size[2]. We also found some previous work when using synthetic data for training purposes, in particular this synthetic data was used to optimize the weights of a model-based reasoning neural network [3], we find this paper really interesting as it deals with using the synthetic data for training purposes, something we would like to research into in future work.

We want to show in figure 1.1 how Dataset, Model and Computations have improved for the last years, with the dataset size remaining the same we need to improve our data exploration[4].

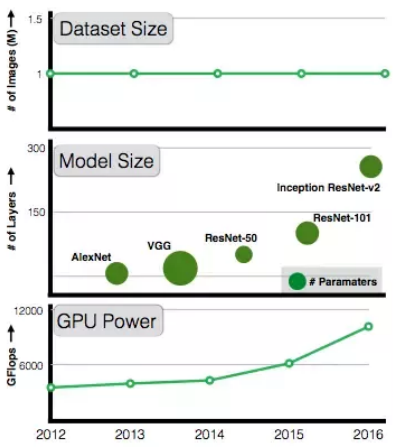


Figure 1.1: Improvements in Data, Models and Computation [1]

With this growing trend companies have also started to research about Machine Learning. At the beginning only big companies such as Google, Amazon or Microsoft were taking profit of it thanks to their resources (computational power, Machine learning experts, big datasets). Datasets are not usually a problem for those companies so we think that this project can be more interesting for the small and medium companies that want to use those techniques but do not count with as much data.

Therefore this project aims to create a tool of data exploration for those situations where dataset size is a problem, the synthetic data will lead to better preprocessing and better train results.

The method we propose to generate synthetic data will analyze the distributions in the data itself and infer them to later on be replicated. We will take special care when replicating the distributions inferred in the data in order to create the most similar data we can.

If this synthetic data is properly created it will be an extra tool when creating models with low event population for both evaluating and understanding them. Finding a way of generating this data can have multiple applications regarding Machine Learning, from just making developing and debugging easier or to make viable the creation of models for datasets with not enough valid data.

1.1 Problem Statement

Machine learning has become more trending in the last decade and so the tools to implement those technique have improved, this improvement is mainly based in more efficient and bigger model creation and more computational power. The purpose of this paper is to research the effect of focusing the efforts in the dataset we deal with and not models or computational power. We aim to answer the following questions: How to create reliable synthetic data given a data collection and does this data reproduces the special features from the original data? and by reliable data we mean data similar to the original.

1.2 Scope

This project studies a reliable method to generate synthetic data given an existing data collection, so that the synthetic data is capable of replicating special features from original data such as special shapes or outliers that makes the original data valuable.

Chapter 2

Background

In this chapter we will explain the key concepts involved in our project. Concepts concerning the data generator will be treated first, we will start with Hidden Markov models and move to the probability theory used in the project. Secondly we will explain the goodness of fit, the measurement we used to choose the distributions we infer from the data. Lastly the technologies and data used will be discussed.

2.1 Hidden Markov Models

A Hidden Markov Models (HMM) is a statistical tool used to represent the probability distributions over sequences of observations when traveling between states[5]. Hidden Markov models will be used in the project to represent the statistical distributions we observed from the data. A Hidden Markov Model consist of :

- N Number of states.
- K Number of events.
- Initial state probabilities,
  $\pi = \pi_i = P(x_1 = i)$ for $1 <= i <= n$
- State-transition probabilities,
  $A = a_{ij} = P(x_t = j | x_{t-1} = i)$ for $1 <= i, J <= n$

- Discrete output probabilities,
  $B = b_i(k) = P(o_t = k|x_t = i)$ for $1 <= i <= n$ and $1 <= k <= n$

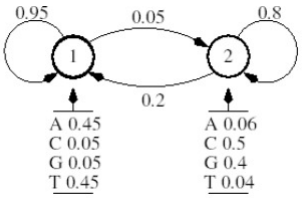


Figure 2.1: Hidden Markov Model representation

In figure 2.1 we find a Hidden Markov Model representation where we can see the parameters previously explained. In this case N is 2 and K is 4, if we look to the state 1 we can find the his State-transition probabilities (0.95 of traveling to state 1 and 0.05 of traveling to state 2) and the discrete-output probabilities.

2.2 Probability distribution

The mathematical definition for a probability distribution in the probability theory and statistics field is a function that provides the probability of occurrence of outcomes in a specific experiment[6]. It satisfies the following properties:

- The probability of x resulting in a specific value is $p(x)$, having $P[X = x] = p(x) = p_x$.
- $p(x)$ is a non-negative value for all real x.
- The sum of $p(x)$ for all the possible values of x is 1.

Probability distributions are usually divided into 2 classes, discrete probability distributions and continuous probability distributions. Discrete probability distributions are those where the set of possible outcomes is discrete , meaning that the distribution can be defined by a discrete list containing the probability for each outcome. Alternatively the continuous probability distribution consist of those where the possible outcome takes values from a continuous range.

Probability distributions can have different sample spaces, distributions with whose space is a set of real numbers are called univariate and the ones with a vector space multivariate. univariate distributions give the probability of one single variable while multivariate distributions gives the probabilities of the vector (joint probability distribution).
One of the key concepts in probability distributions is the probability density function (pdf), a function whose value at any given sample in the sample space (all the possible values for the random variable) can be seen as the relative likelihood of the value of the random variable which would equal that sample.

2.2.1 Dirichlet distribution

The Dirichlet distribution (usually denoted as $Dir(\alpha)$ is a multivariate distribution that describes $K >= 2$ variables $X_1, ..., X_k$ such that for every $x_i \in (0, 1)$ and $\sum_{i=1}^{N} x_i = 1$, those X are parameterized as a vector $\alpha = (\alpha_1, ..., \alpha_k)$ [7]. We define the probability density function for $Dir(\alpha)$ as:

$$\frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

where

$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}$$

Dirichlet distributions are mostly used as the prior distribution for categorical variables in Bayesian mixture models or other hierarchical Bayesian models. It is also known as the a multivariate generalization of the beta distribution. Figure 2.2 presents an example of a Dirichlet probability density function along α

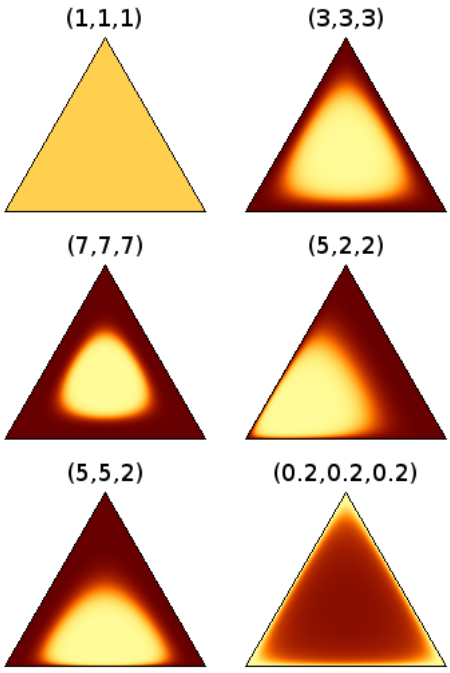
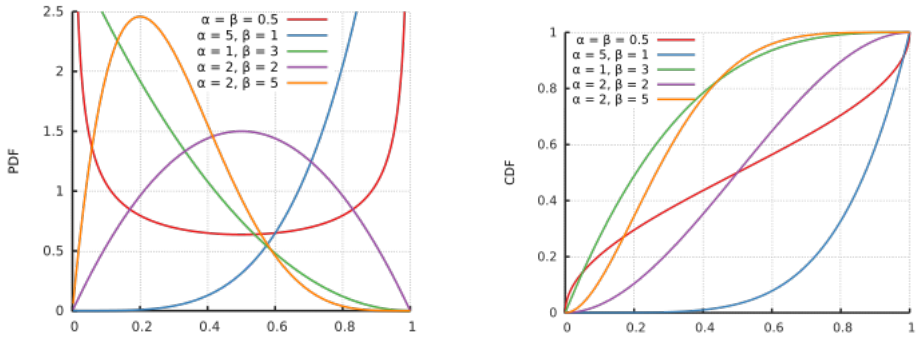


Figure 2.2: Example of a Dirichlet probability density function along α

2.2.2 Beta distribution

The Beta distribution is a continuous probability function defined on the interval $[0, 1]$ with parameters α and β [8]. These two parameters act as exponents of the random variable and also shape the distribution. The probability density function of the beta distribution:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{(\alpha,\beta)}$$



(a) Beta probability density function

(b) Beta Cumulative distribution function

Figure 2.3: Beta pdf and cdf

In figure 2.3 we can see how beta distribution pdf and cdf and also how it shapes as a uniform distribution when $\alpha = \beta = 1$.

2.3 Goodness of fit

The goodness of fit for a statistical model describes how well does this statistical model fits for a set of observations[9]. In other words, it tells you if your sample data represents the data you can expect from that statistical model. Some of the goodness of fit test commonly used in statistics:

- The chi-square.
- Kolmogorov-Smirnov.
- Anderson-Darling.
- Shipiro-Wilk.

2.3.1 The chi-square

This test is used for discrete distributions as binomial or Poisson, alternatively the Kolmogorov-Smirnov can only be used for continuous distributions. The chi-square distribution is a special case of the gamma distribution and is one of the most widely used probability distributions in inferential statistics.

Chi-square test can only be used for labeled data and it usually requires a lot of samples for the approximation to be valid. The formula for the chi-square test:

$$\tilde{\chi}^2 = \frac{1}{d} \sum_{k=1}^{n} \frac{(O_k - E_k)^2}{E_k}$$

2.3.2 Kolmogorov-Smirnov

Kolmogorov-Smirnov test (K-S test) does not actually tell you if a sample comes from a specific probability distribution, instead it tells if you can reject the hypothesis of that data coming from the probability distribution so it does not make assumptions about the distribution of the data.The Kolmogorov-Smirnov Test is based on the cumulative distribution function of the underlying distribution[10].

Samples can be compared to distributions using one-sample K–S test (against a probability distribution and his parameters) or two-sample K–S test (against other set of different samples).

Being $F(x)$ the probability distribution we are given and the empirical distribution function F_n for n observations X_i we define the empirical distribution function $F_n(x$ as:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} \begin{cases} 1, & \text{if } y_i <= x, \\ 0, & \text{otherwise} \end{cases}$$

We also define the Kolmogorov–Smirnov test statistic as:

- For K-S 1 sample: $D_n = sup_x |F_n(x) - F(x)$.
- For K-S 2 samples: $D_{n,n} = sup_x |F_{1,n}(x) - F_{2,m}(x)$.

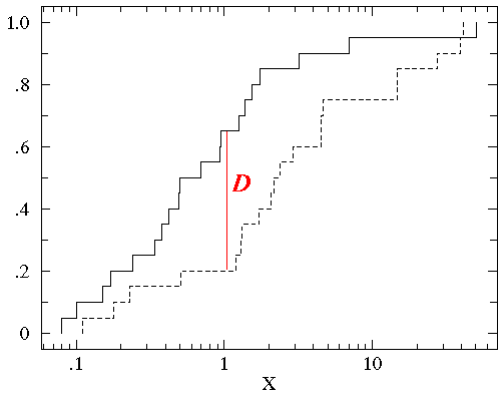


Figure 2.4: KS test comparison cumulative plot

In the figure 2.4 we can see the Kolmogorov–Smirnov test statistic between two different samples.

2.3.3 Anderson-Darling

This test is a modification from the previous Kolmogorov-Smirnov test, being more sensitive to deviations in the distribution tail we are testing against. It works as the Kolmogorov-Smirnov test, it will tell you when it is unlikely that your data comes from the distribution you are comparing to. The formula for this test:

$$S = \sum_{k=1}^{N} \frac{2k-1}{N} \left[\ln F(Y_k) + \ln\left(1 - F(Y_{N+1-k})\right)\right]$$

2.3.4 Shapiro-Wilk

This test is slightly different from the previous ones, Shapiro-Wilk returns a value W that tell us if the sample we are testing comes from a normal distribution. This test only works for normal distributions unlike the other tests.

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{(\sum_{i=1}^{n} x_i - x)^2}$$

2.4 Technologies and data used in the project

This project was entirely coded in python, several data processing libraries were used such as: Numpy, Pandas, Scipy. Numpy and pandas libraries were responsible for the data pipelines and Scipy for the distributions inferring. The generating method was entirely coded from scratch.

Regarding the data we will be using data from telecom nodes, we selected the tables containing the throughput of internet connections that go through a node. This data is time-stamped and contains 59 variable, we collected more than 120 millions of samples with a size of 60gb. Scaling by Minmax has been applied to this data to improve our results.

Chapter 3

Method

In this chapter we want to give a deep and clear explanation of how our model works. We propose a new method for generating synthetic data, remember that this method pursues synthetic data capable of replicating special features from the real data. The main advantage of this method when replicating data is the structure of states with different probability distributions, making it capable of replicating those special features by capturing them in states.

3.1 Creating the model

To create our model we will analyze the statistical distributions followed by the given data and estimate their parameters. Then a Hidden Markov Model will be created where each state will contain the parameters for the statistical distribution inferred from the data collection. More than one state can be created in the same Hidden Markov Model as we can encounter multiple distributions inside a single variable and by identifying them we increase the ability to replicate special features from the data. We present now an example of what this model consist of.

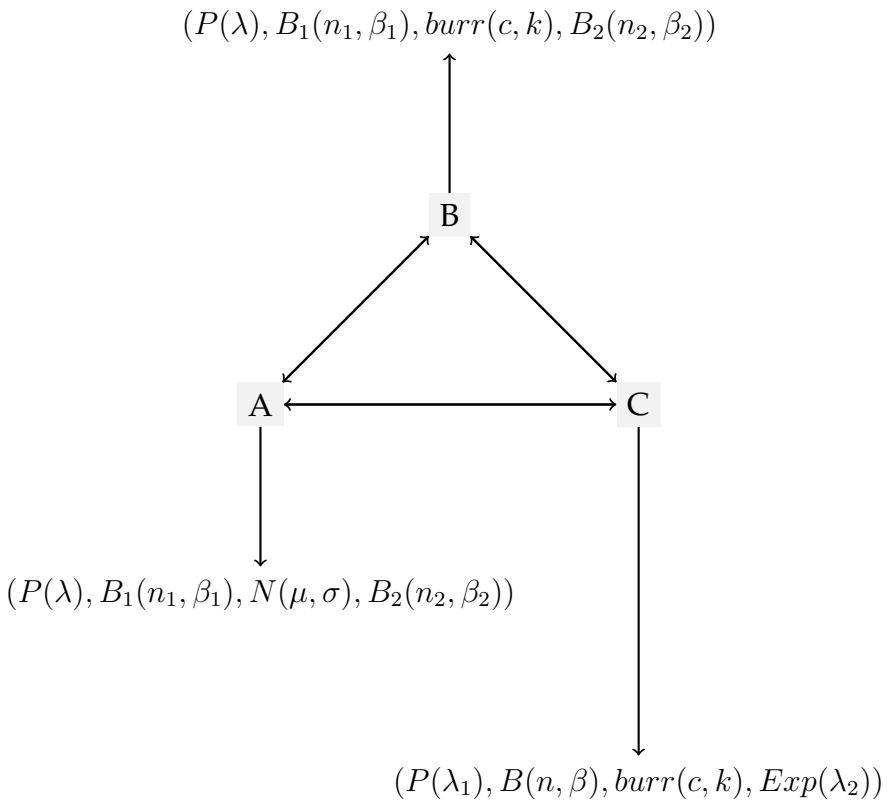


Figure 3.1: Representation example of our Hidden Markov Model

The figure 3.1 is a representation of our Hidden Markov Model for a dataset with 4 variables. In this example we have 3 states A, B, C and each one of them contains a list of statistical distributions parameters. One example for this parameter distributions would be: $P(\lambda_1)$ a Poisson distribution. In this model the dataset contains 4 variables so each state contains 4 statistical distribution parameters. By the definition of our model states can not have the same list of statistical distributions parameters so we find different list for each state, each of those list represents an specific behaviour of our data that we have captured with the statistical distributions inferred. An example of this special behaviour would be the third variable for the dataset that in B and C follows a $burr(c, k)$ distribution while in A is a $N(\mu, \sigma)$ distribution.

To create this Hidden Markov model we first need to figure out how many states we need to represent our data and also infer the statistical distributions for those states, we explain this in the following section.

3.1.1 Probabilistic distribution parameters inferring

We need the inferring to be as precise as possible as conversely it will lead to non similar data when we move to the generation process. We will first infer from all the data collection to later infer different windows in the data, this will allow us to find different distributions in the same data and create more than one state for the HMM.

When performing the inferring for the given data we will fit the data variables one by one for a set of both Discrete and Continuous distributions. To ensure we choose the distribution that fits better our data two test will be performed. The first test will be performing a Goodness of fit test with Chi-square test used for discrete variables and Kolmogorov-smirnov test for Continuous variables. The second test will consist of creating a probabilistic data function for the each most likely distributions and compare it with an histogram of the variable, we will measure the sum of square error and choose the distribution with the lower one. With this two test we will chose the distribution that replicates our data in the best way, we will then store its parameters.

Once the inferring for all the data is done a first state will be created, containing a finite number of distribution parameters equal to the number of variable in the data. For the windowing inferring we will check if the distributions we obtain represent a difference with the first state, if 20 percent of the distributions are different a new state will be created.

3.2 Generating data

We want the algorithm to travel between states to be smooth and non drastic so this project presents an algorithm to travel between states where a sample is created in each step and traveling requires multiple steps. Traveling between states takes S (number of total states of the model or ten if $Nstates < 10$)steps to perform a change from state $S1$ to $S2$ and work as follows.

First we create a Dirichlet distribution $Dir(\alpha)$ where α parameters represent each state percentage of occurrence, α parameters will be normalized. This will be the prior distribution to be in each state.

To start generating data we choose a state $S1$ to start from the $Dir(\alpha)$ prior distributions and create a sample of data from the dis-

tribution parameters contained in it. In the next step we will choose again a state $S2$ to travel from $Dir(\alpha)$ prior distributions, if $S2 = S1$ we will stay in the current state and sample again from the distribution parameters contained in it, otherwise we will start traveling from $S1$ to $S2$ where S consecutive steps pointing $S2$ are required to change to $S2$ and finish the traveling. If while traveling between states $S1$ and $S2$ a state $S3$ is chose as destination state we will stop the traveling process and start again from $S1$. While in this traveling between states the data generated will be a weighted sum of data generated from the distributions from the starting state and the destination state, this weighted sum will take in account how many consecutive steps traveling to the destination we have done so far.

Chapter 4

Results

This chapter describes the results retrieved from the experiment with their respective discussion. A total of 12 millions samples where used to create the model. Each sample contains 59 continuous parameters. After the data inferring we created synthetic data with our HMM model. Here we present charts comparing the real data with the data we created, remember that this project seeks to replicate data that maintains the original data features. The measure rate for the distributions we infer from the data is the P value we obtain from the Kolmogorov–Smirnov test or Chi square test.

4.1 Overview and discussion

4.1.1 Fitting distributions

We present now charts with some of the variables with inferred from where we compare the histogram of the data with the distributions pdf we fit the data to. In the first chart we present all the possible distributions we were capable to fit for the variable and in the second the distribution pdf of the top distribution in terms of goodness of fit.

Figures for the statistical distribution inferring of variable $throughput\_downlink$.

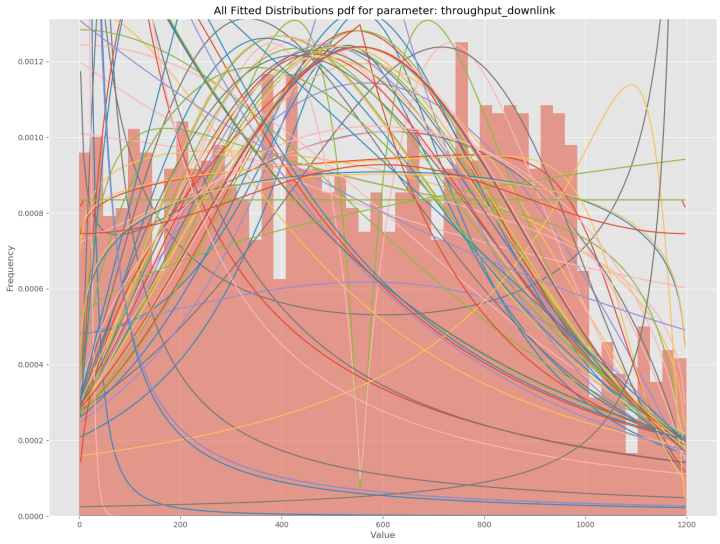


Figure 4.1: All pdf from distributions fitted to our data, variable $throughput\_downlink$.

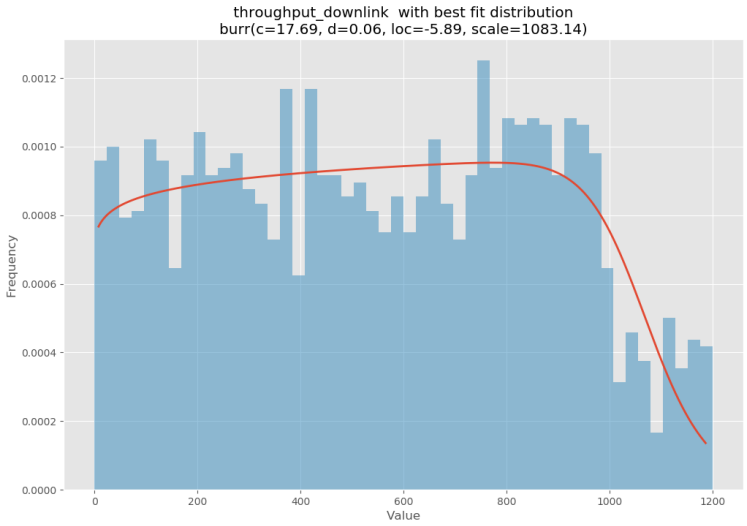


Figure 4.2: Pdf from the distribution $burr(c = 17.69, d = 0.06, loc = -5.89, scale = 1083.14)$ with best goodness of fit P=0.45, variable $throughput\_downlink$.

First we see the fitting for the variable $throughput\_downlink$ whose figure 4.1 represents all of those pdf from the distribution we fitted to our data. Between all of those pdf we choose the one for the $burr$ distribution in figure 4.2. This burr distribution obtains the best result in the goodness of fit Kolmogorov–Smirnov test where P=0.45.

Figures for the statistical distribution inferring of variable mbr_uplink.

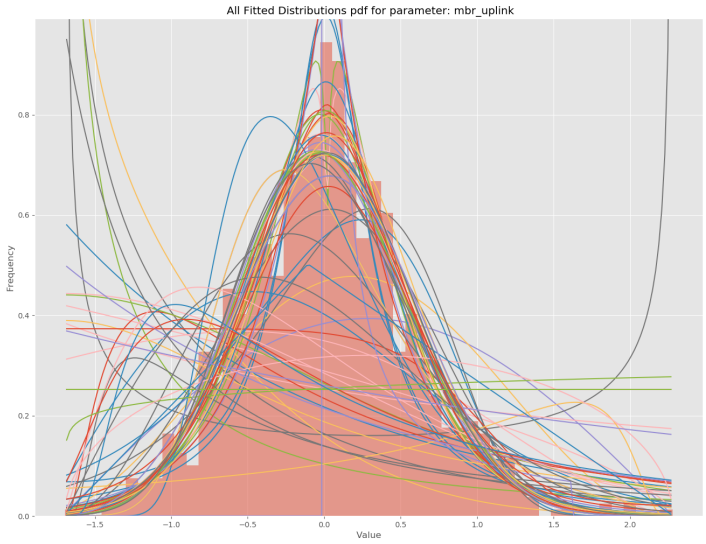


Figure 4.3: All pdf from distributions fitted to our data, variable mbr_uplink.

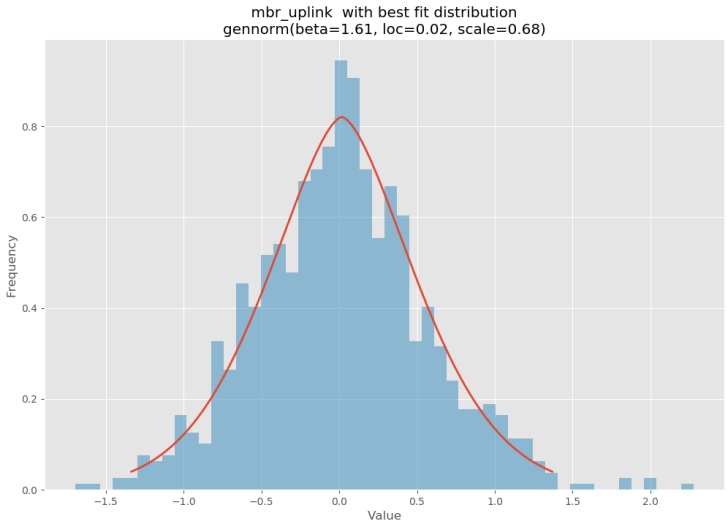


Figure 4.4: Pdf from the distribution $gennorm(beta = 1.61, loc = 0.02, scale = 0.68)$ with best goodness of fit P=0.77, variable mbr_uplink.

Moving on to the next variable mbr_uplink we see how in figure 4.3 there are a lot of pdf that are very similar to our data, meaning that our resultant pdf will have a high score in the goodness of fit test. In figure 4.4 we see the resultant pdf that comes from a Generalized normal distribution with a result of P=0.77, if we compare this result with the previous variable $throughput\_downlink$ and his resultant pdf with P=0.45 we see how having a pdf more likely to our data translates in a higher goodness of fit result.

Figures for the statistical distribution inferring of variable $peak\_throughput\_uplink$.

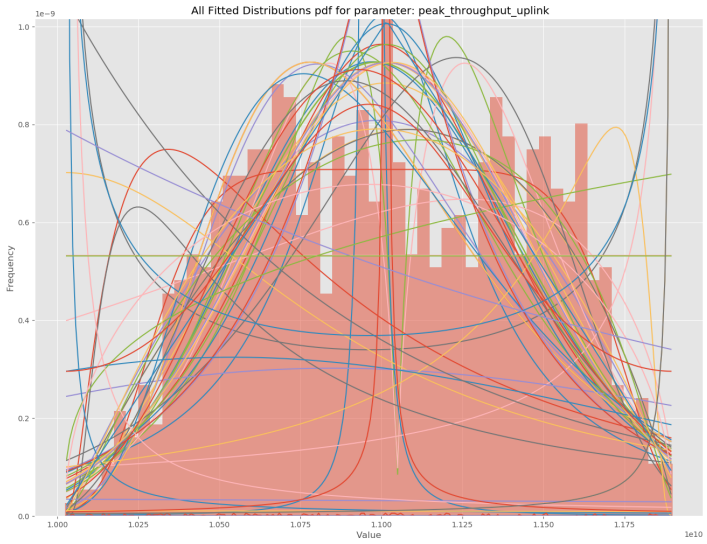


Figure 4.5: All pdf from distributions fitted to our data, variable $peak\_throughput\_uplink$.

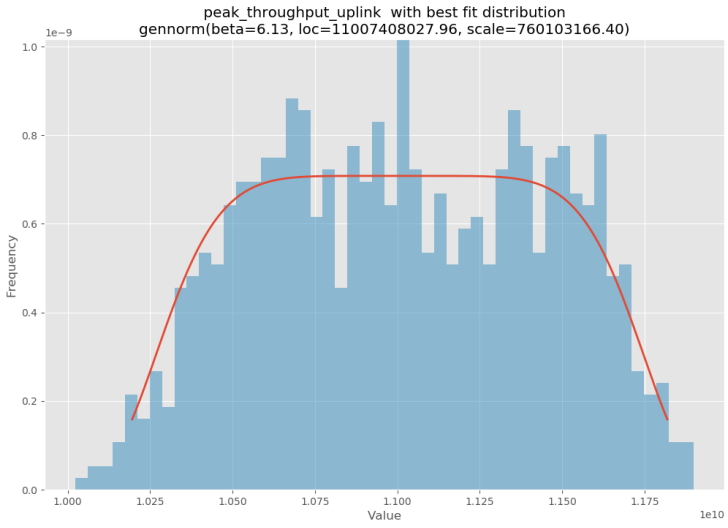


Figure 4.6: Pdf from the distribution $gennorm(beta = 6.13, loc = 11007408027, scale = 760103166.40)$ with best goodness of fit P=0.68, variable $peak\_throughput\_uplink$.

By last we analyze the $peak\_throughput\_uplink$, with this variable we find that the pdf from the distributions we fit are more similar than in $throughput\_downlink$ but less that in mbr_uplink. The resultant pdf comes from a Generalized normal distribution (beta=0.63, loc=11007408027.96, scale=760103166.40) as in mbr_uplink but with a lower goodness of fit result P=0.68, this lower result compared with mbr_uplink comes from the gap in the histogram between values 1075 and 1150, if there would not have any gap there the resultant distribution would probably have been a normal distribution.

4.1.2 Generating data

Now we present charts comparing the real data against the synthetic data.

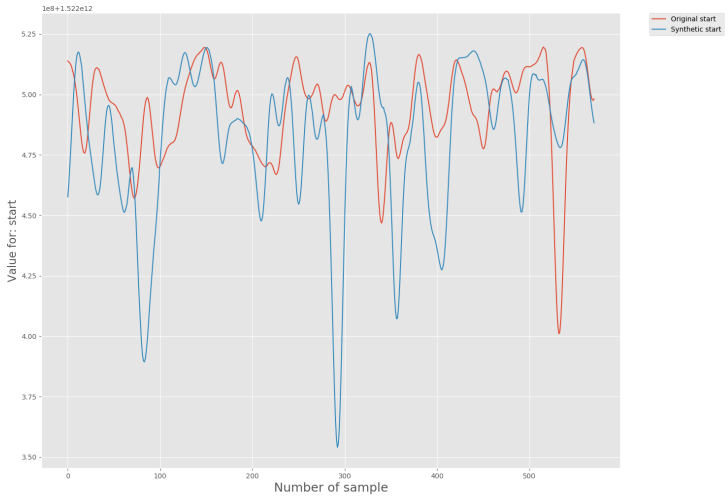


Figure 4.7: Real data against synthetic data for the variable $star$.

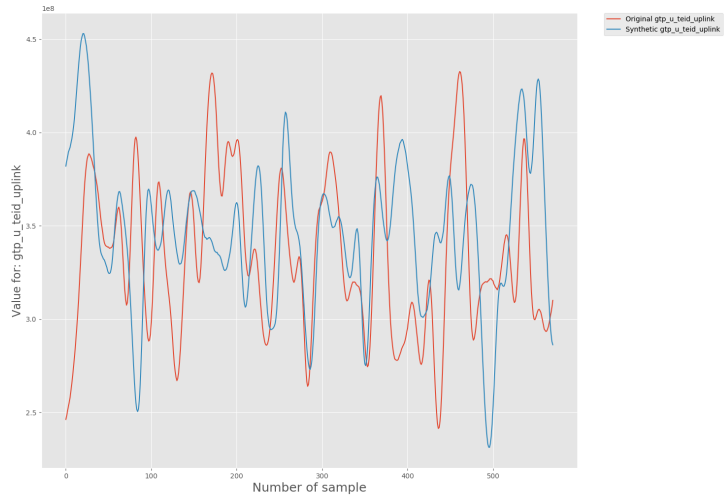


Figure 4.8: Real data against synthetic data for the variable $gtp\_u\_teid\_uplink$.

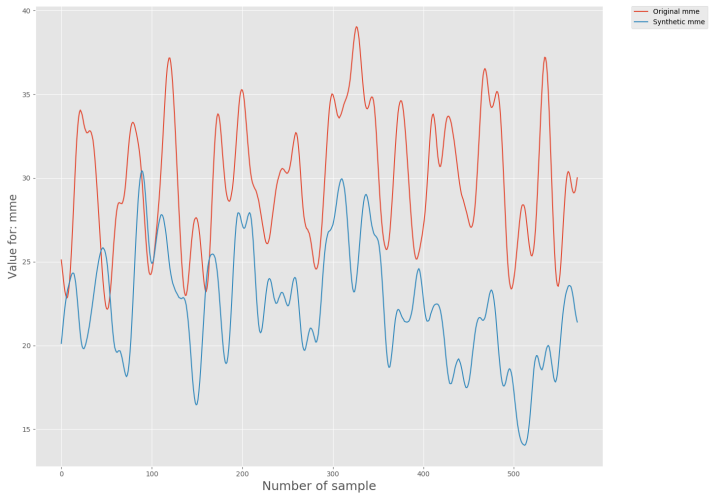


Figure 4.9: Real data against synthetic data for the variable mme.

Moving on to results with synthetic data generation we find the synthetic data generated for different variables, first we find the variable $start$ where we see that our inferred distributions create big significant outliers (sample 90 and 290) which does not replicate the real data. We think that this outliers are due to the real outlier in the data (sample 550). Our model is not capable of isolating this single outlier and this translates in several synthetic outliers that do not correspond with the data.

Next synthetic data created comes from $gtp\_u\_teid\_uplink$ and seems to have better results than the previous $start$, here the real data does not have any big outlier and is more consistent, the spikes in the data are something usual. Here the synthetic data does a better job that with the previous variable, those spikes are correctly replicated and we can see how the data is nearly similar to the real one.

The final synthetic data to analyze is the one for mme, here the real data and synthetic data are not as similar as in the previous examples, the synthetic data is slightly biased to lower values but it still keeps the shape from the real data. We see how synthetic data replicates the spikes in samples 50-150, 200-300,300-370. We think this bias is due to the lower values from the real data, in $gtp\_u\_teid\_uplink$ we had more similar synthetic data but values where in the order of 10^8, here our

values are in the interval $15 - 40$. This leads us to the hypothesis that our model is more capable of replicating data when higher values are treated but his ability to replicate data shape remains the same despite the value size.

So far the synthetic data we have presented comes from the method where we transit between states, now we present data generated from the state with less prior probability in our Dirichlet distribution, there is no state traveling.

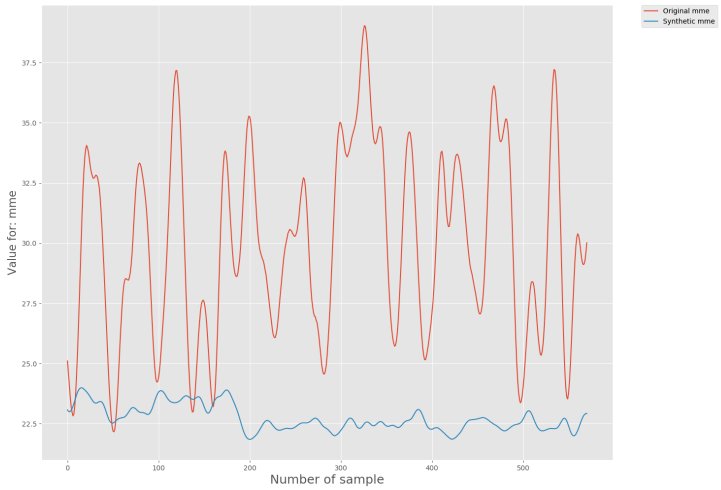
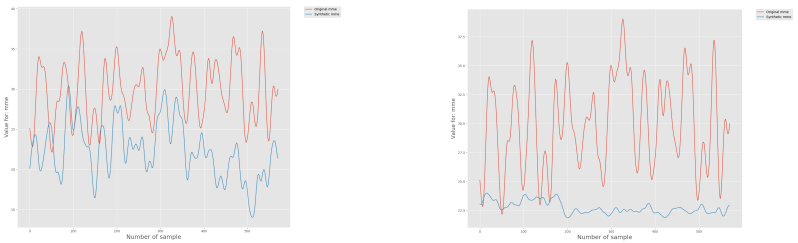


Figure 4.10: Real data against synthetic data for the variable mme, data generated from the state with less prior probability.

Figure 4.11: Real data against synthetic data for the variable mmc, data generated from the state with less prior probability.

This data is generated with the statistical distributions contained in the state with less prior probability in our Dirichlet distribution so we were not expecting any reliable results, our expectations were confirmed. The first figure 4.10 we see the variable mee the same as in figure 4.9 (data generated for the same variable but from all the states). We compare them now side by side:



(a) label 1 (b) label 2

Figure 4.12: mme data from all the states against mme data from the state with less prior probability.

We see how the synthetic data in figure 4.12 (a) is more similar than 4.12 (b), figure b corresponds with the data from the state with less

prior probability, meaning that is the distribution less likely to simulate real data. We think this state reproduces the lower outliers from the real data.

Figure 4.11 represents data generated from the state with less prior probability for the variable mnc. We can see how the synthetic data that comes from this state does not reproduce the real data at all, the probability of this state was by the order of -10^{15}. This state is useless in terms of generating data similar to the real one, it probably was created due an outlier.

Chapter 5

Discussion

This chapter discuss the possible improvements, limitations and future work for the project.

5.1 Limitations

This project was mostly limited by the tremendous amount of cpu power required to infer data distributions from such a big dataset as we had (12 millions samples). The inferring computations were handled by the library Scipy for python and the lack of a gpu support really slowed us during the process. More states could have been explored if the computations would have been faster.

We also encounter some limitations when dealing with such a big dataset. A pipeline was needed to handle all the data transferring along the model, limiting the time used for perfecting the model.

5.2 Ethics and sustainability

When we think about the ability to replicate data there is an infinity of purposes we can use it for. One possible use case would be medical data such as heart rate histograms. Synthetic data allows us to fulfill data gaps for a tremendous number of use cases. It helps with the data exploration allowing the user to understand the data better and identify special features of the data. Generating synthetic data also helps us to get a view of how a bigger dataset would be, this view could save us from actually getting a bigger dataset and all the consequences and

work necessary. In the same line the synthetic data allows us to know if a model would be useful with our data by providing early results with the synthetic data, giving us a performance preview without the need of fetching more real data.

5.3 Possible improvements and future work

As stated in the result discussion our model lacks of efficiency when generating data with a small amount of big outliers. Removing these outliers is not an option as we are not aiming to reproduce the most of the data we possibly can, we aim to reproduce special shapes or features (such as this outliers) from the data while reproducing the majority of the data. An interesting improvement for our model would be to improve the ability of replicating data when dealing with such outliers. Implementing the distribution inferring with gpu support would improve our computation time and allow us to spend more time looking for the right distribution.

We also plan to test this model with other kinds of datasets, explore how our model behaves when dealing with other datasets will help us to test and improve it. In this dataset we only had a few categorical variables so the inferring was mostly for continuous variables. Another possible improvement would be to be able to infer multivariate data distributions, right now only univariate distributions are tested and the addition of multivariate could improve our model efficiency when replicating data by extracting more accurate distributions from it. In summary we now have demonstrated a reliable way to create synthetic data but only for one dataset, we need to test this method in other datasets to see how in behaves.

Chapter 6

Conclusion

The result of this study showed that the model we propose is able to create data capable of simulating the real one but that there are still some improvements that could perfect the model. Additionally and to answer our research question, the data we create its similar to the original one and can also replicate those special features in some cases, there is still room for improvements in this special features replication. In this study we have seen how we can replicate one dataset but that does not prove it will work with other kinds of datasets so more datasets need to be tested to have a better perspective of this method results. By last we want to remark how the method to travel between states is able to provide smooth data creation.

Bibliography

[1] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era", *CoRR*, vol. abs/1707.02968, 2017. arXiv: 1707.02968. [Online]. Available: http://arxiv.org/abs/1707.02968.

[2] J. Shotton, A. Fitzgibbon, A. Blake, A. Kipman, M. Finocchio, B. Moore, and T. Sharp, "Real-time human pose recognition in parts from a single depth image", IEEE, Jun. 2011. [Online]. Available: https://www.microsoft.com/en-us/research/publication/real-time-human-pose-recognition-in-parts-from-a-single-depth-image/.

[3] R. Z. Tuan Anh Le Atılım Gunes¸ Baydin and F. Wood, *Using synthetic data to train neural networks is model-based reasoning*, 2017. DOI: https://arxiv.org/pdf/1703.00868.pdf.

[4] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data", vol. 24, pp. 8–12, May 2009.

[5] S. R. Eddy, "Hidden markov models", *Current Opinion in Structural Biology*, vol. 6, no. 3, pp. 361–365, 1996, ISSN: 0959-440X. DOI: https://doi.org/10.1016/S0959-440X(96)80056-X. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959440X9680056X.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006, ISBN: 0387310738.

[7] S. K. N. B. N. L. Johnson, *Continuous Multivariate Distributions Volume 1: Models and Applications*. Wiley, 2000, ch. Chapter 49: Dirichlet and Inverted Dirichlet Distributions, ISBN: 0-471-18387-3.

[8] N. Johnson, S. Kotz, and N. Balakrishnan, *Continuous univariate distributions*, ser. Wiley series in probability and mathematical statistics: Applied probability and statistics v. 2. Wiley & Sons, 1995, ch. Chapter 21:Beta distributions, ISBN: 9780471584940. [Online]. Available: https://books.google.se/books?id=0QzvAAAAMAAJ.

[9] Q. Liu, J. Lee, and M. Jordan, "A kernelized stein discrepancy for goodness-of-fit tests", in *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., ser. Proceedings of Machine Learning Research, vol. 48, New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 276–284. [Online]. Available: http://proceedings.mlr.press/v48/liub16.html.

[10] M. A. Stephens, "Edf statistics for goodness of fit and some comparisons", *Journal of the American Statistical Association*, vol. 69, no. 347, pp. 730–737, 1974. DOI: 10.1080/01621459.1974.10480196. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1974.10480196. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1974.10480196.